Detect Cancer in Pathology Images

Applied Deep Learning - Spring 2019 Course Project

Amal Alabdulkarim - aa4235

Overview

- Each year, the treatment decisions for more than 230,000 breast cancer patients in the U.S. depends on whether the cancer has metastasized away from the breast.

- This is determined by performing a Sentinel node biopsy. During this procedure the surgeon removes the sentinel nodes and send them to the pathologists to analyze them in a laboratory.

- This task is very important but requires large amounts of reading time from pathologists. Therefore, there is a need for a developing tools to assist the pathologists and reduce their workload while at the same time reduce the subjectivity in diagnosis.

Solution Overview

- Two Multi-scale prediction models with a fine tuned inception V3 base and Resnet50 base.
- Inception V3 base model uses a similar approach to **“Detecting Cancer Metastases on Gigapixel Pathology Images”** with several differences, such as:
    - Use Zoom level 2 as the lowest zoom level in the model instead of 40x.
    - Use Zoom level 3 as the high zoom level in the model instead of 20x.
    - Use Adam instead of RMSProp optimization method.
- Inception V3 model performed well on the paper therefore, it is a good model for this task and can be used to predict cancer in pathology images.
- Resnet50 enhances the detection of smaller objects in the image. Tumor markers are smaller objects, that may be missed, that is why I wanted to try using Resnet50 for this task, to see what can we learn from the Residuals.

Dataset

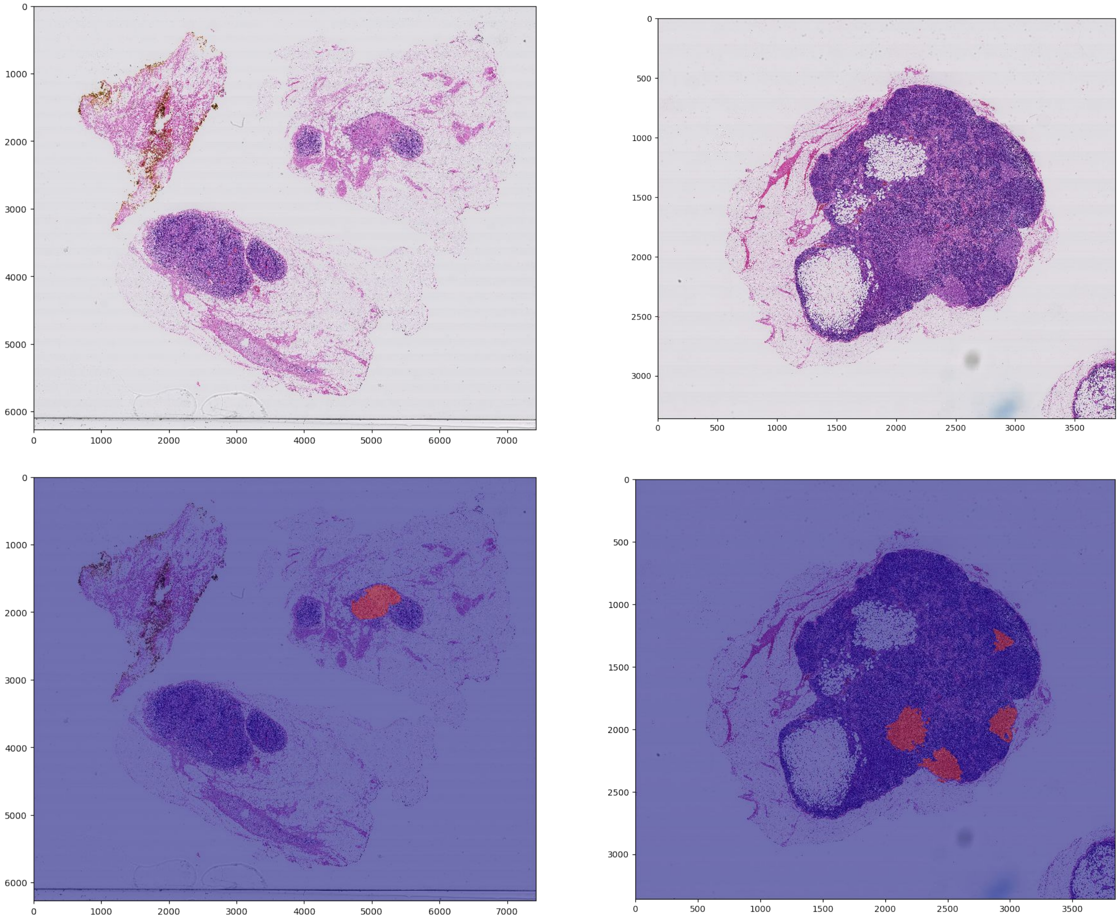
- CAMELYON16 challenge dataset.
- A subset of 22 slides has been provided in the Google Drive folder.
- At first I ran a validation script to filter the slides, the remaining slides satisfies two criterias:
  - It passes the dimension verification, which ensures that the dimensions of the mask are equal to the dimensions of the slide at all zoom levels.
  - It passes the downsampling verification, which ensures that the downsampling it correct.
- This validation left me with 7 slides, and then I manually inspect them and removed 2 slides that had a very small tumor area to an additional test set, because it would increase the training data imbalance and I can't train on more than 4 slides due to computational limitation.

Dataset
Training, Validation and Testing

- 4 training and validation and 3 for testing.
- Randomly selected 1 image for testing using sklearn train test split function (0.2 split). (And added the other two manually)
- Randomly selected 20% of the training patches for validation using Keras ImageDataGenerator validation split and flow_from_directory subset parameter.

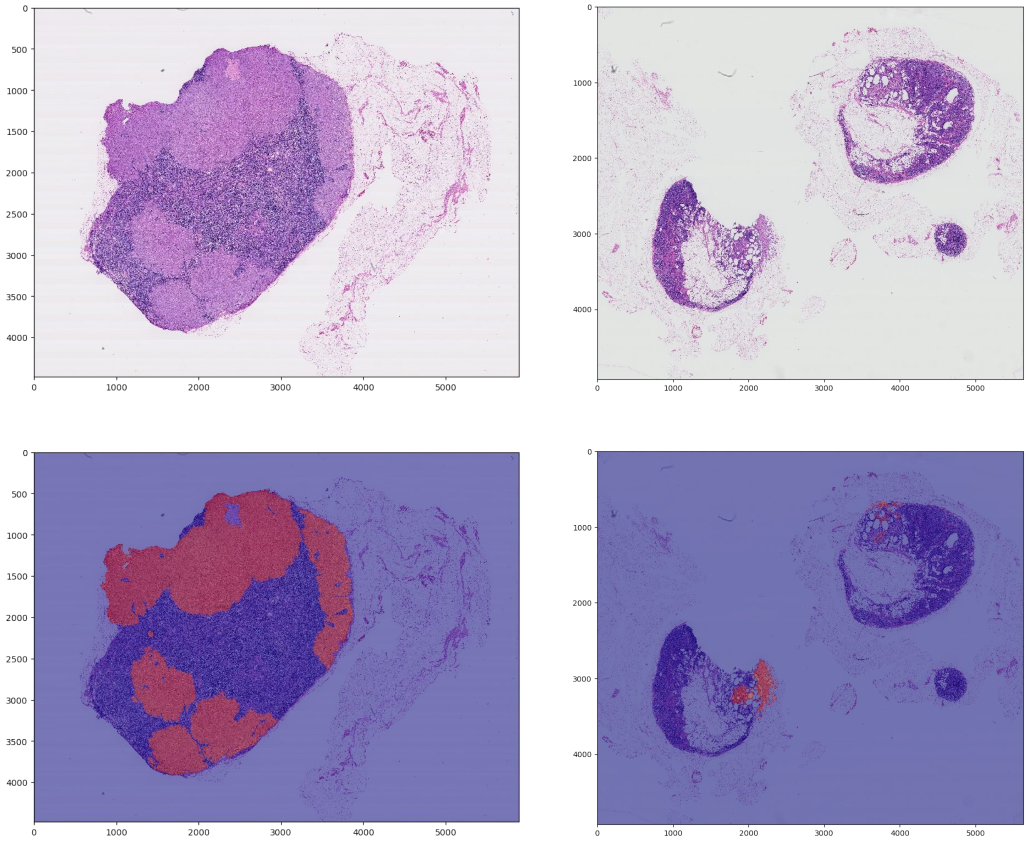
Dataset

Training Set

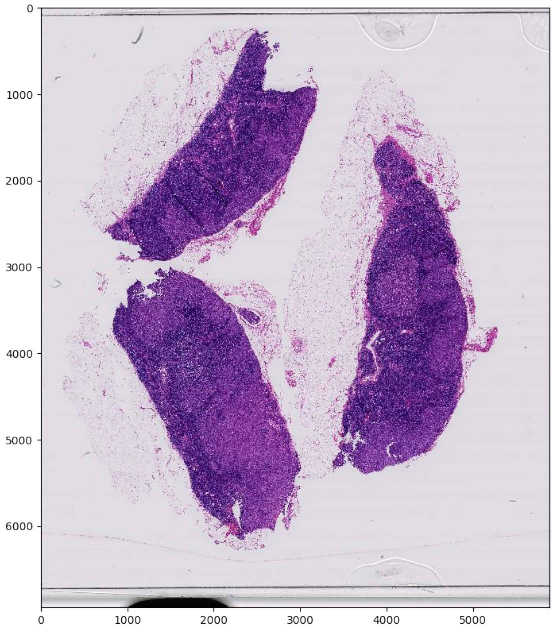


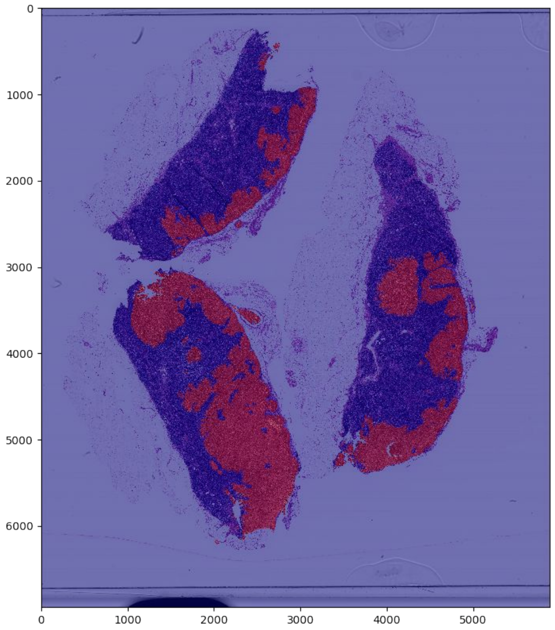
Dataset

Training Set

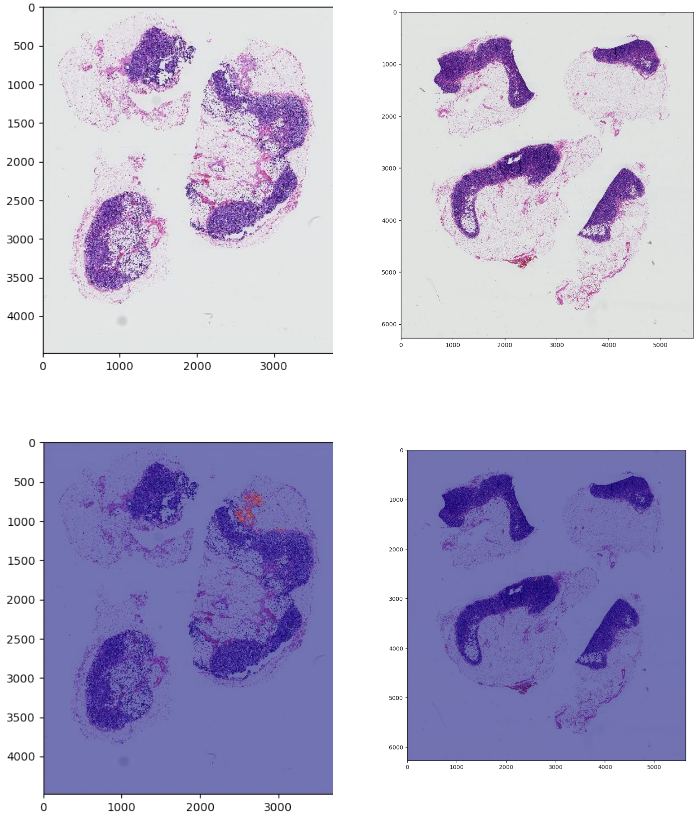


Dataset
Test Set





Dataset
Additional Test Set



Preprocessing

1. Data Augmentations:
   a. Horizontal Flip.
   b. Vertical Flip.
   c. 90 degrees rotation.

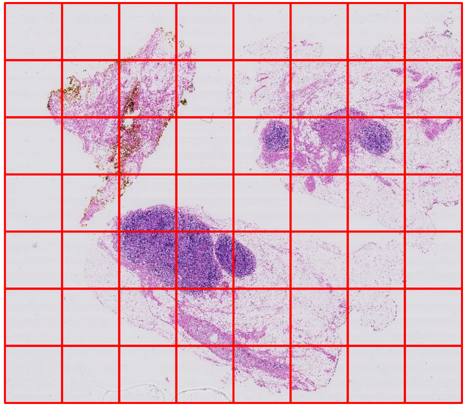
2. Normalization.
   a. Divide each color value by 255 to make its range (0-1) instead of (0-255). This step improves computational efficiency of the model. Match RGB values in the pretrained inception model.

Patch Generation and Labeling

1. Each slide was divided into 299*299 pixels patches, in levels 3 and 2.
2. Each patch was labeled by counting tumor pixels at the 128*128 area in the level 2 patch. If this area contains at least one tumor pixel, it is classified as tumor.

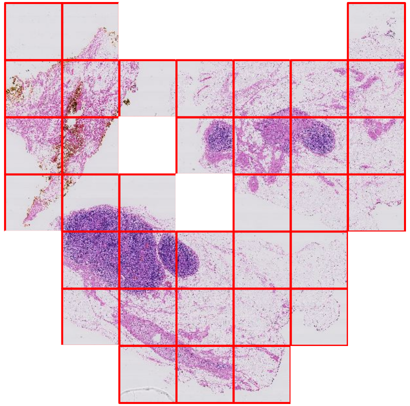
Preprocessing Steps

Extract Patches with more than 20% tissue from each slide



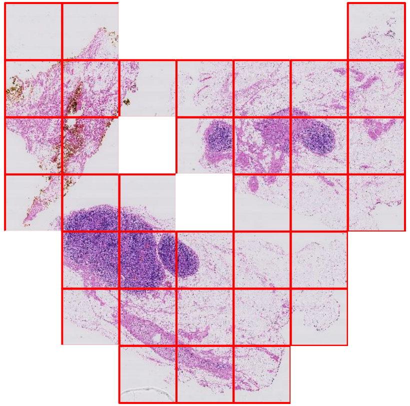
Preprocessing Steps

Extract Patches with more than 20% tissue from each slide

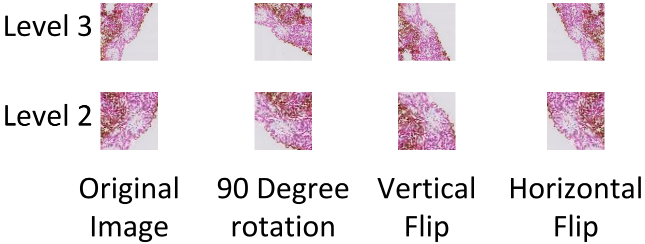


Preprocessing Steps

Extract Patches with more than 20% tissue from each slide

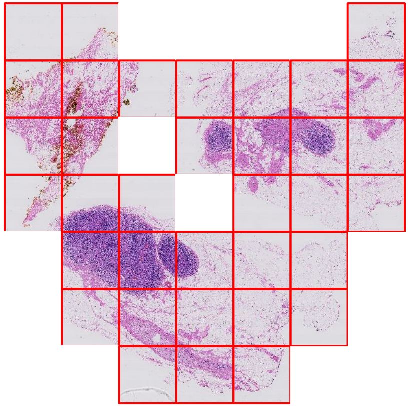


Generate lower zoom level sample and augment patch.



Preprocessing Steps

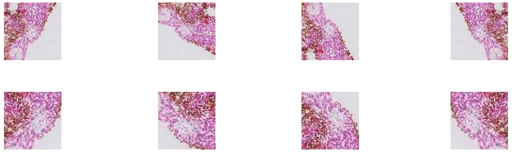
Extract Patches with more than 20% tissue from each slide



Generate lower zoom level sample and augment patch.

Level 3

Level 2



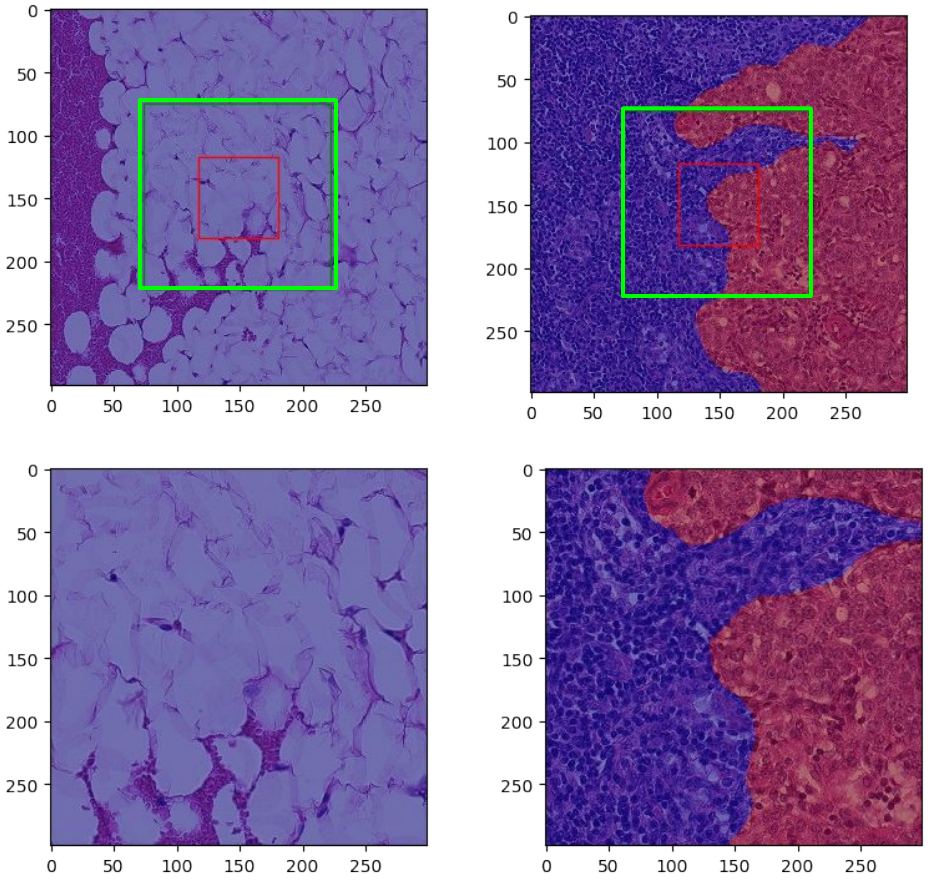
Original Image | 90 Degree rotation | Vertical Flip | Horizontal Flip

Generate Label From the center 128*128 of the lower level

Patch Label = 0

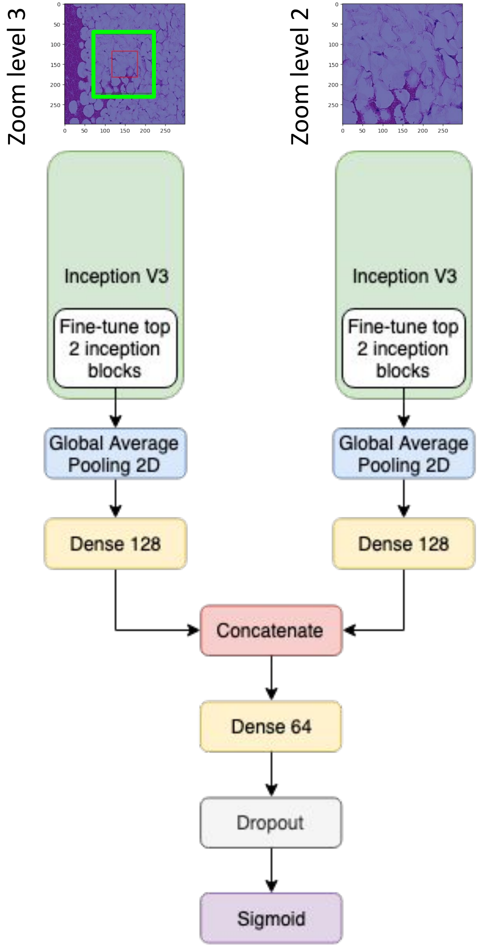
Patch Generation and Labeling

Examples of a negative patch and a positive patch



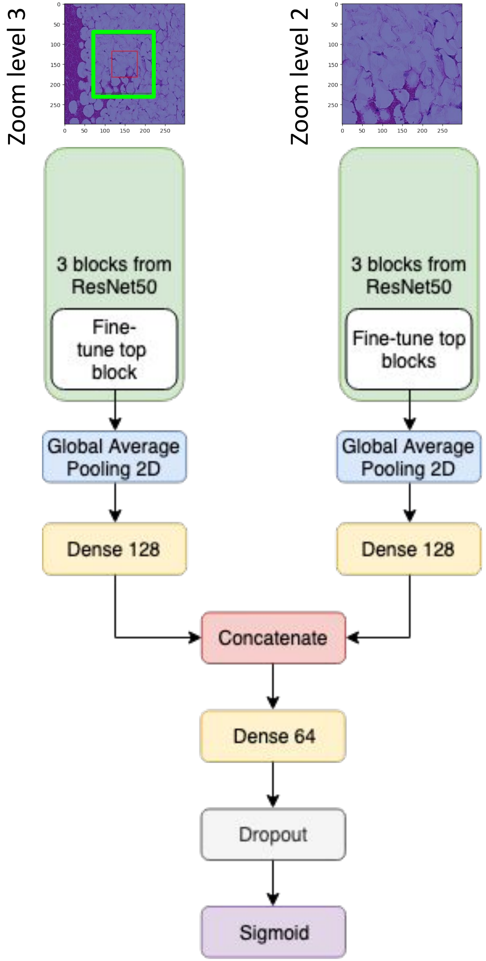
Inception V3 Model

1. Two inception V3 towers, each one takes as an input a zoom level centered image.
2. Inception V3 model is initialized using imagenet weights.
3. The top 2 inception blocks are fine tuned and trained with the model.
4. The three dense layers have “relu” activation.
5. Dropout rate is 0.2



Resnet50 Model

1. Two Resnet50 towers, each one takes as an input a zoom level centered image.
2. Resnet50 model is initialized using imagenet weights.
3. Only the first 3 blocks from the model are used, the first two blocks are frozen, the last third block is trained with the model.
4. The three dense layers have “relu” activation.
5. Dropout rate is 0.2



Training Parameters

1. Binary Cross Entropy loss.
2. Adam optimizer with 3e-4 learning rate.
3. Batch Size = 6
4. Epochs = 30
5. Callbacks on improving validation accuracy.

Balancing data

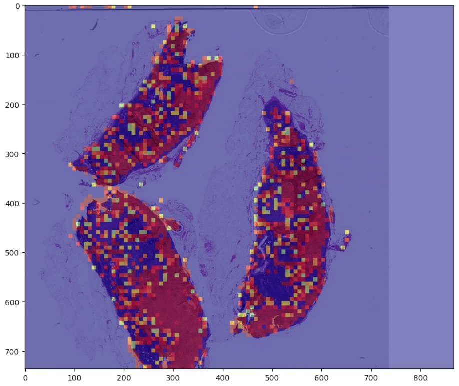
- To balance training and validation data I built a custom generator to sample batches from the directory, and then balance these batches by random sampling (with replacement) to make sure that each batch contains even instances of each class.
- This method will upsample the minority class.

Results

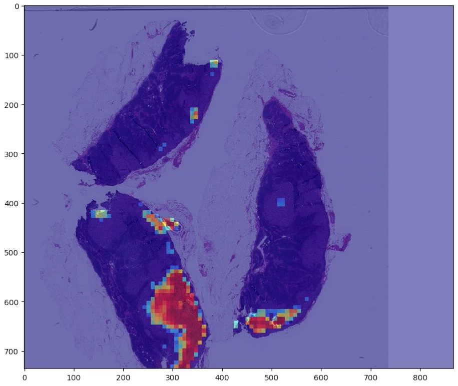
Predicted Label (threshold > 0.5)

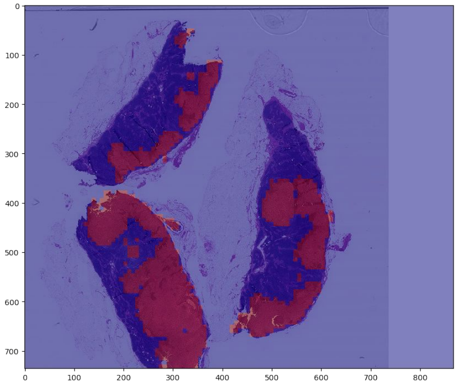
True Label

Inception V3



Resnet50

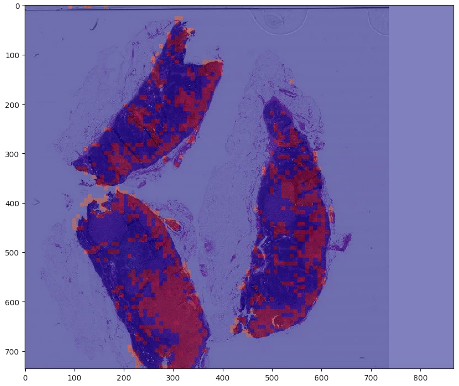




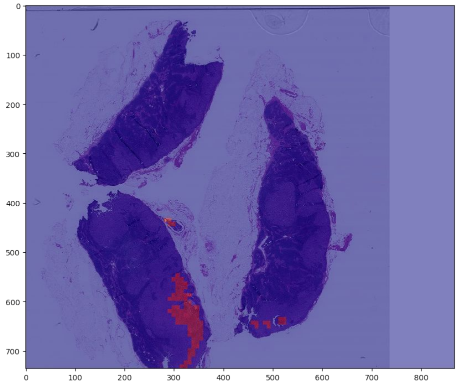
Results

Predicted Label (threshold > 0.95)

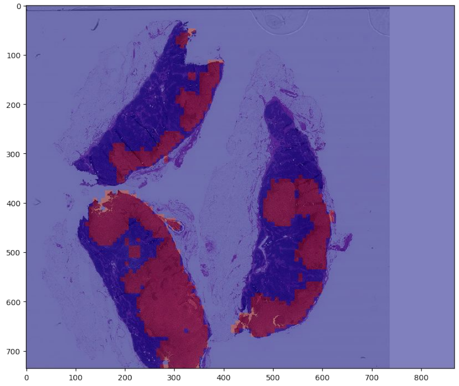
Inception V3



Resnet50



True Label

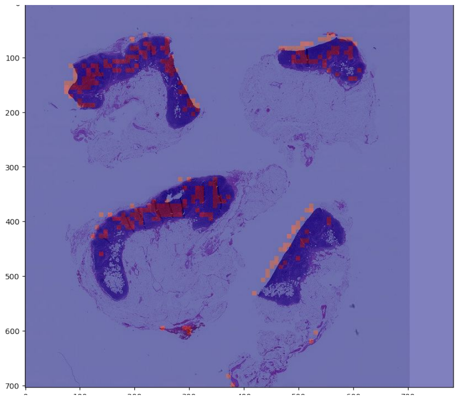


Results

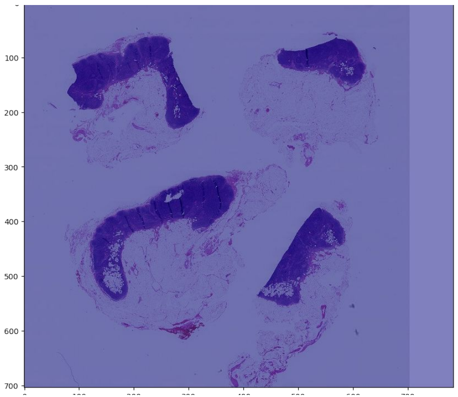
Predicted Label (threshold > 0.95)

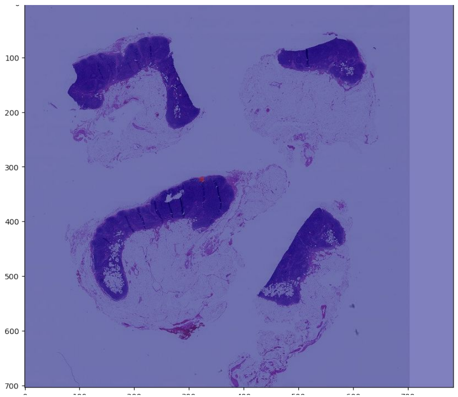
True Label

Inception V3



Resnet50



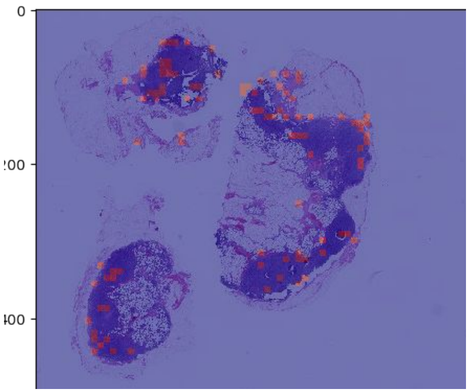


Results

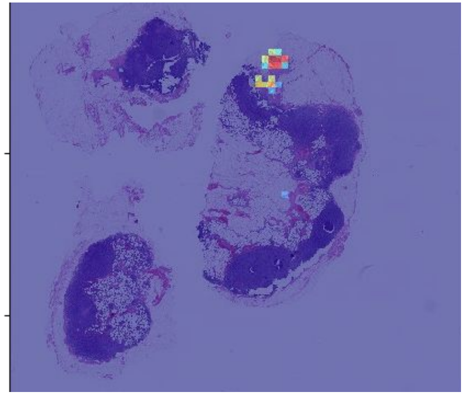
Predicted Label (threshold > 0.95)

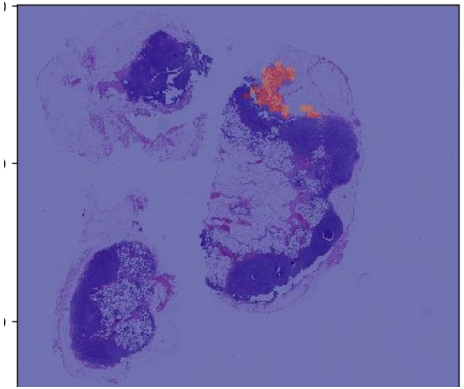
True Label

Inception V3

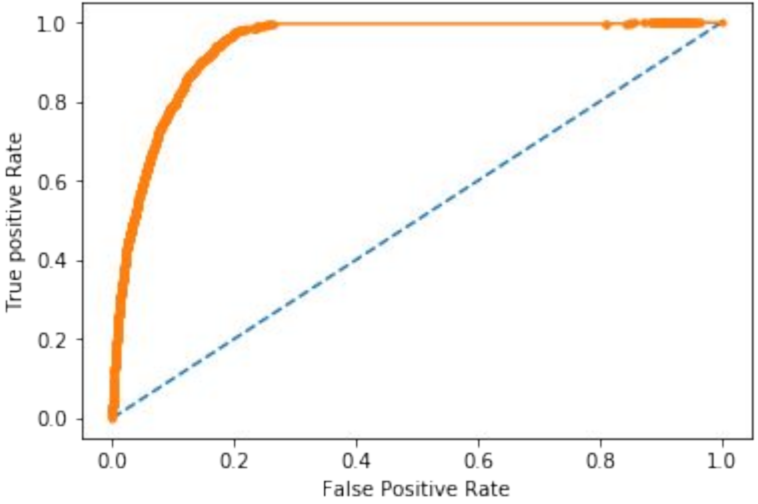


Resnet50

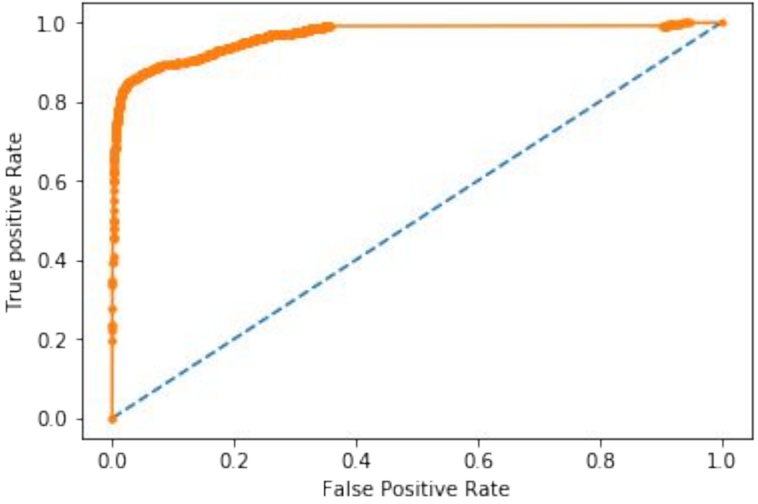




Evaluation



Inception V3 AUC score: `0.942`



ResNet50 AUC score: `0.967`

https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html

Evaluation

Inception V3 Balanced Accuracy score: `0.87`

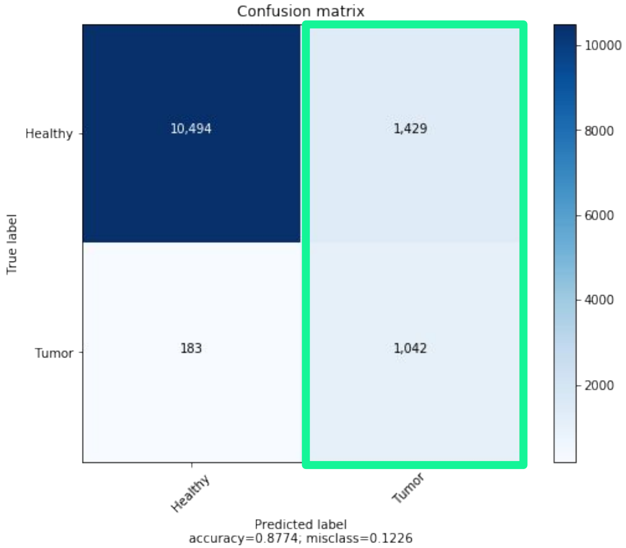
Resnet50 Balanced Accuracy score: `0.60`

$$\texttt{balanced-accuracy}(y, \hat{y}, w) = \frac{1}{\sum \hat{w}_i} \sum_i 1(\hat{y}_i = y_i)\hat{w}_i$$

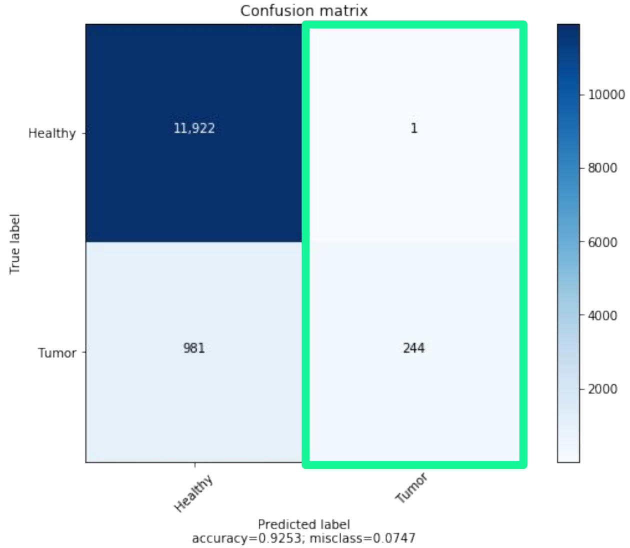
https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score

Evaluation

Inception V3 Confusion Matrix



Resnet50 Confusion Matrix



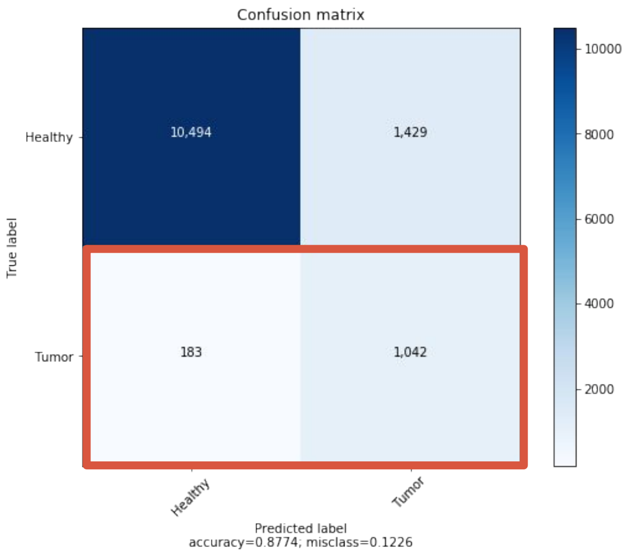
Inception V3 Precision: 0.42

ResNet50 Precision: 0.99

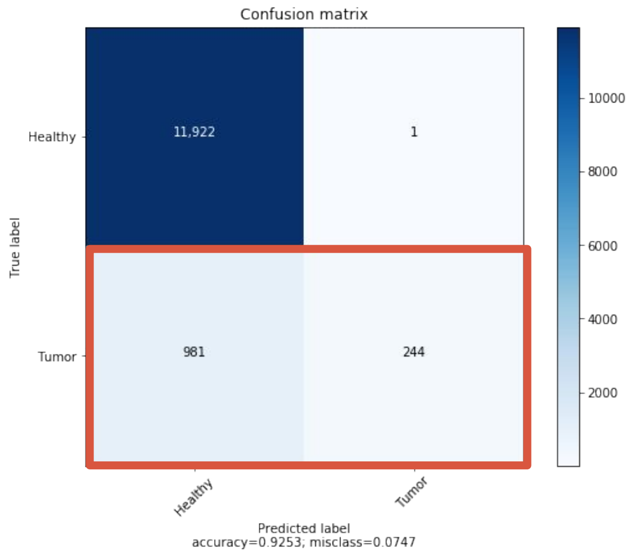
http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Evaluation

Inception V3 Confusion Matrix



Resnet50 Confusion Matrix



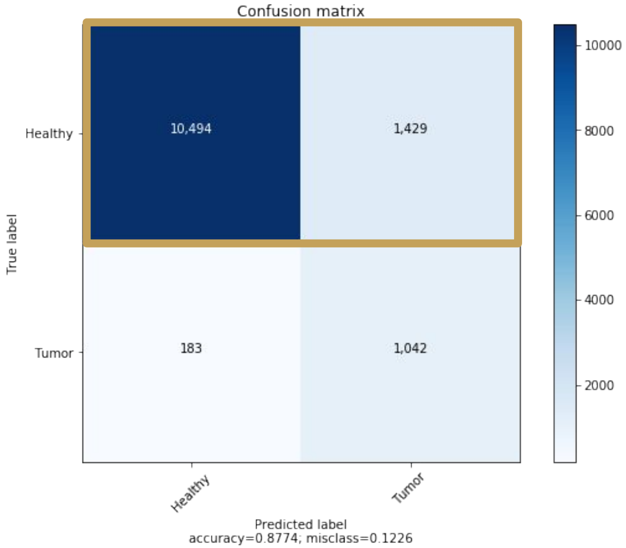
Inception V3 Recall/Sensitivity: 0 . 85

ResNet50 Recall/Sensitivity: 0 . 20

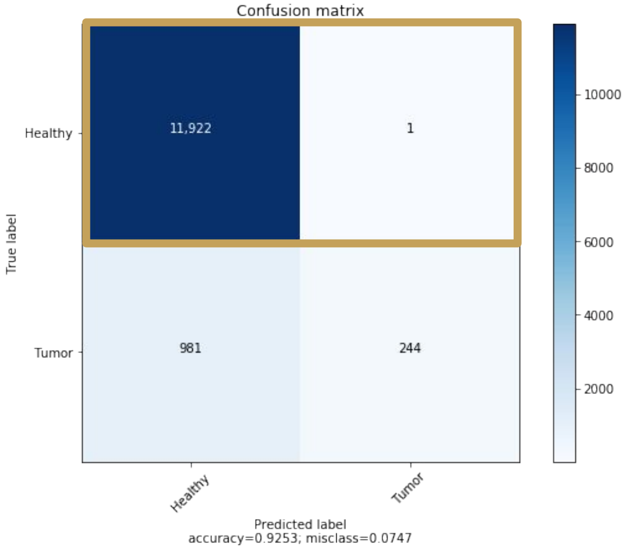
http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Evaluation

Inception V3 Confusion Matrix



Resnet50 Confusion Matrix



Inception V3 Specificity: 0 . 88

ResNet50 Specificity: 0 . 99

http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Conclusion and Discussion

- Both prototype models have high AUC scores which suggest that an improved model can be used as a tool to assist pathologists in detecting cancer.
- Precision, sensitivity and specificity are all very important for evaluating the models' predictions. There is usually a tradeoff between those metrics and domain experts can help in determining what should be more important to the model.
- The ResNet model has a very high precision and low sensitivity and the Inception V3 model has a high sensitivity and low precision.

Future Work

- Visualize the models activations and try to understand what does each model learn.
- Train the model on more images and using lower zoom levels.
- Retrain inception V3 on a large tumor dataset instead of using imagenet weights.
- Try more data augmentation methods.
- Utilize other pretrained models on medical data.